Patterns of Communication Behaviors among Global Software Student Teams and the Effects of Task Type

Kathleen Swigger (corresponding author)
Department of Computer Science & Engineering
University of North Texas
1155 Union Circle #311366
Denton, TX 76203-5017
phone: 940-565-2817
kathy@cs.unt.edu

Fatma Cemile Serce
Atilim University
Ankara, Turkey
06836 Incek Golbasi Ankara Turkey
phone: +90 312 586 83 75
cemileserce@gmail.com

Ferda Nur Alpaslan
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
+90-(312)-210-5544
alpaslan@ceng.metu.edu

Robert Brazile
Department of Computer Science & Engineering
University of North Texas
1155 Union Circle #311366
Denton, TX 76203-5017
brazile@cs.unt.edu

George Dafoulas
T123, Town Hall
School of Engineering & Information Sciences
Middlesex University
The Burroughs, Hendon London, NW4 4BT
phone: 020 8411 4402
G.Dafoulas@mdx.ac.uk

Victor Lopez
Universidad Tecnológica de Panamá
Panama City, Panama
Apartado Postal: 0819-07289,
Panamá, Republic de Panamá
victor.lopez@utp.ac.pa

Patterns of Communication Behaviors among Global Software Student Teams and the Effects of Task Type

Abstract

A key factor in the success of global software development learning teams is the communication that occurs among the group. Various task characteristics, however, may affect the both the quality and quantity of the group communication. This study investigates the effects of task type on the communication behaviors of student teams engaged in a software development project. Two groups of teams completed assignments that varied in degree of task type and product. Content analysis was used to identify distinct patterns of interactions and examine how these patterns were associated with task type. Results indicate that differences in task context and product do not have large effects on the communication behaviors of global software teams. These findings will provide a basis for creating instruction that can help maximize successful communication among global software learning teams.

Keywords: Global software development, Computer Supported Collaborative Learning, Distributed Work Groups, Teamwork

1. Introduction

Changes within the software industry have prompted computer science educators to develop new courses that teach students how to work in global software development teams [5, 13]. A critical component of these courses is the completion of a software project that requires students to work in multi-university teams that cross time as well as geographical boundaries [2, 29]. The internationalization of software engineering courses has been particularly useful because they can bring a degree of realism to the classroom and make learning more relevant. They also provide unique opportunities for students to learn how to communicate knowledge within the context of a culturally mixed distributed team [4, 18]. Similar to real-work situations, students must learn how to overcome obstacles such as differences in culture, time zones, and languages [9, 30, 33] in order to produce a final product. While new collaborative technologies, such as shared management tools and mature videoconferencing systems, seem to be helping students communicate across time and space, many questions remain about how to teach students to share ideas, knowledge and code. Lacking specific instructional materials that teach students how to interact more effectively with their team members, instructors have tended to rely on more experiential learning activities to deepen students' understanding of the group communication process. Evidence suggests, however, that student-to-student interactions alone do not always lead to better performance among global software development learners [8, 26]. The need for research about how to make global software student teams more effective has prompted questions about which practices and training lead to better performance. As the geographic scope widens, educators are striving to understand the challenges and opportunities introduced by globally distributed courses in order to provide a more competitive software education for their students.

To examine issues related to the teaching of global software development courses, the authors began a multi-university research project that is aimed at increasing the effectiveness of distributed programming teams that are composed of students who have different cultures and live in different time zones. One of the major objectives of the research is to develop instructional materials that help students use new technology to communicate and share ideas, code, and information. The specific universities involved in the research project are Middlesex University (MDX), Universidad Tecnológica de Panamá (UTP), University of North Texas (UNT), Middle East Technical University (METU), and Atilim University (AU). Each semester, students from the participating universities are grouped together and asked to complete a software development project. The classroom projects are intended to mimic the inherent Global Software Development (GSD) characteristics of geographical distance, different cultures, and different time zones. Using various computer-supported collaborative tools, students must learn how to communicate with their teammates and coordinate the different software development tasks. Because these interactions are recorded, we are able to examine the different communication activities in an effort to determine which factors lead to better performance. These particular analyses are designed to give us useful insights into the specific dynamics that affect distributed teams. They also provide a basis for selecting strategies that can either maximize or minimize the various factors that characterize more successful collaborations.

One of the results from a study completed in spring 2008 suggested that communication patterns might be related to task type [37]. Student teams located in the United States (US) and the United Kingdom (UK) were assigned a database project, while student groups in the US, Panama and Turkey were given a programming exercise. An analysis of students' online discussions indicated that there was a significant difference in the communication behaviors of the US-UK groups versus those in the US-Panama-Turkey projects. The groups in the US-Panama-Turkey project displayed significantly more contributing and less planning behaviors than the groups in the first project. The difference in these two behaviors, we believed, was due to differences in the tasks assigned to the two groups of students. The UK-US teams were asked to design and query a database, while the US-Panama-Turkey student groups were assigned a programming task in which the individual teams in each country had to produce a component of a larger program. It was speculated that the latter task probably necessitated a greater amount of contributing behaviors such as giving feedback, etc. Thus, particular types of tasks may generate specific communication patterns among global software development student teams.

The research presented in this paper is an attempt to investigate whether aspects of the task affect the communication patterns of global software learning teams. The implication for such a study is that the results should help teachers of global software development courses provide their students with more informed information about how to communicate with distributed team members. Similarly, the authors sought to examine the affect of task type on the communication patterns of high versus low performing teams. In order to address these issues, the authors analyzed the computer conferencing transcripts of two different global software development student projects by means of a content classification scheme developed by Curtis and Lawson [10].We then examined the results of these classifications to compare the distribution of communication behaviors that occurred within each project, and whether there were any differences between groups based on task type or performance. The paper begins with a report on the relevant research that was used to guide this study, followed by an overview of the experiment. The paper also includes a description of the coding scheme and the measures that were used to gather data about individuals and teams. Finally, the paper presents the results of our analyses and concludes with a list of recommendations that are meant to improve future work.

2. Related Literature

Communication seems to be an essential component of all software development collaboration practices and processes. Besides formal project communication, empirical studies suggest that developers rely heavily on informal, ad hoc communication [25, 32, 39]. Consequently, hurdles in communication can have dramatic effects upon team members' abilities to complete global software development (GSD) projects. Besides differences in language and culture, global software development teams suffer from a lack of informal communication, resulting in low levels of trust and awareness of work and progress at remote sites. In GSD projects, managing communications is important. Strategies recommended in the literature include the use of special communication liaisons [17], bridgehead teams [6], and technical personnel to help interpret different communication styles and patterns among team members.

Communication also plays an important part in the success (or failure) of distributed learners. There are numerous studies that support the idea that interactions with both the instructor and other students are essential ingredients in distributed learning courses [14, 38]. For example, Garrison, Anderson, and Archer [15] describe the importance of creating a "virtual community of inquiry" that allows learners to construct experiences and knowledge through analysis of the subject matter, questioning, and challenging assumptions. The importance of communication is probably even more critical for global software student teams, given that in such teams, computer-mediated communication forms the basis of all social action [34] and knowledge transfer [43]. It has been argued, for example, that a student who engages in a higher extent (or greater amount) of communication will transfer more knowledge to his/her remote team members, thus leading to better team performance. As a result, performance on a project or assignment is often measured by looking at the number of chats or notes posted by a student. Similarly, studies have used the number of online activities such as the number of messages [19], mean number of words [2], and thread-length [19] to assess the extent of student collaboration.

It is now widely believed that reporting on the quantity of communication activities alone is not sufficient to understance group collaboration [27]. To understand the true effects of a particular communication activity, researchers suggest using content analysis to assess the quality of online discussions [12]. Content analysis allows researchers to discover the existence of certain patterns in online discussions and determine how particular patterns affect the performance of a group [40, 41]. A communication pattern is usually established through the use of a particular coding scheme that characterizes an online interaction. For example, reference 42 classifies student messages into three categories: (1) added, explained, or evaluated; (2) summarized; (3) transformed. This scheme can be seen as an information processing approach because each activity represents a different level of information processing. On the other hand, Walther [44] describes communication patterns in terms of personal, interpersonal and hyper-personal behaviors. Still other educators have developed coding schemes that describe students' critical thinking skills, which are then used to measure the quanity of such activities within an online discussion [28, 31]. Coding schemes have also been developed for determining the overall meanings of a set of postings, and how these different meanings are transferred to a participant's ability to perform other related tasks [7, 16, 36]. Finally, researchers such as Jeong [22] and Bakeman [3] analyze the entire discussion in an attempt to learn about the relationships and transitions that occur within and among different interactions. Comparison of content analysis instruments used to characterize student discussions reveals that classification schemes often vary according to particular tasks or activities assigned to students [24]. For example, examining student discussions about different themes in literature may require the researcher to use a different type of coding scheme than looking at how student groups solve math problems.

2.1. Task Products

Several researchers have suggested that students' discussion often varies with the specific task that is required. Wiley and Bailey [45] point out that successful group work occurs when students must work together to achieve a specific goal (i.e., accomplish an interdependent task). It is likely that a specific end product stimulates students to share information in a particular way and to discuss and learn from the knowledge that other students bring to the mutual task [42]. Andriessen [1] suggests that the groups often focus on the specific themes and problems of the discussion task. Specification of a specific product, therefore, may affect the way students communicate and how they share ideas, knowledge and information.

On the other hand, several researchers have found that task differences were less strong in online communications. For example, Hollingshead [20] found that the relationship between technology and task performance appeared to be more dependent on experience with the technology and with group membership than on the type of task. It has been suggested that variables such as gender [35], knowledge [21], and medium may have more effect on communication patterns than task or goal.

2.2. Research questions

This study investigates the effect of task context on the communication behaviors present in asynchronous online discussions for global software learning teams. This study also seeks to assess the relationship between task type and patterns of communication among high and low performing teams in asynchronous online text discussions. The following hypotheses were investigated:

- *H_1: The communication patterns of global software development teams assigned Task 1 and those assigned Task 2 are the same (homogenous).*

- *H_2: The communication patterns of high performing global software development teams assigned Task1 and high performing teams assigned Task 2 are the same (homogeneous).*

- *H_3: The communication patterns of low performing global software development teams assigned Task1 and low performing teams assigned Task 2 are the same (homogeneous).*

The measurement approach adopted for this study uses a coding scheme developed by Curtis and Lawson [10], which is a content analysis technique that is used to characterize collaborative communication activities. Curtis and Lawson [10] first identified different types of behaviors (as described in Johnson & Johnson [23]) as being supportive of the collaborative process, and then developed a coding schema that matched these processes to utterances in on-line collaboration. The authors define five categories of collaborative behaviors displayed in messages: (1)

planning, (2) contributing, (3) seeking input, (4) reflecting and monitoring, (4) interacting socially. Individual codes are assigned to postings that indicate specific types of behavior. The authors of this paper used the Curtis and Lawson instrument to place students' discussion messages into various categories of behavior and then examined the communication patterns for the two projects. A more detailed description of the procedures and measures used in the study now follows.

3. Methodology:

3.1. Background

Every semester, the researchers at the five universities collaborate on the design and implementation of a proposed global software development project. The proposed projects generally involve junior or senior computer science or IT students who have completed both an introductory and advanced programming course. The global software development projects also tend to vary according to the skill levels of the participating students and on the specific courses involved in the research study. The two learning objectives that guide the development of the group projects are: (1) students should learn about the challenges and opportunities of collaboration within a virtual setting, and (2) students should gain experience working with people from a different country or culture.

Once the instructors agree upon the assignment, then the students are brought into the process. After being trained on the different types of software, students are introduced to their team members (either through a teleconference or synchronous chat), and are provided information about the task as well as management of the teams. The student teams are asked to use only designated collaborative software to communicate with one another. The various collaborative software systems that are used in the projects support asynchronous communication tools such as forums, emails, file sharing etc., as well as synchronous communication tools such as chat. Since these systems have record keeping capabilities, we are able to capture the communication behaviors for each team.

Students enrolled in these courses generally receive between 10-15 percent credit as part of their overall course grade for completing the project. To further motivate participation, students are also given prizes for their involvement and performance.

3.2. Subjects

A total of 155 students participated in the two global software development student projects that are described in this paper. The participants in the first global student project (Task 1) contained both undergraduate and graduate students; 27 master's level students enrolled in a human factors course at the University of North Texas. 32 students enrolled in a Java programming course at the Atilim University, and 26 students from Universidad Tecnológica de Panamá, all of whom were recruited from different project-oriented courses.

A total of 70 students participated in the second global software project (Task 2). The 10 students from Universidad Tecnológica de Panamá were enrolled in a database course, the 34 students from Atilim University were enrolled in a Java course, and the 26 students from the University of North Texas were enrolled in a database course.

Table 1 summarizes the demographic information for the two projects. Table 1 shows that male students predominated both projects (115 total), but the courses did include some females (40 total). All of the participating students were currently enrolled in a computer science or information technology department.

Table 1. Demographic Information of Subjects

Task 1				
University	**#students**	**Level**	**Male**	**Female**
AU	32	BS	18	14
PTU	26	BS	21	5
UNT	27	MS	18	9
Total	85		57	28
Task 2				
AU	34	BS	26	8
PTU	10	BS	7	3
UNT	26	BS	25	1
Total	70		58	12

AU: Atilim University, Turkey
PTU: Universidad Tecnológica de Panamá, Panama
UNT: University of North Texas, US

For the first task, the average grade point average (GPA) for students in Panama and Turkey was around 2.0, while US students averaged 3.6 (which would be expected given that some of these participants were graduate students). For the second task, the average GPAs for Panamanian and Turkish students was around 2.5, while US students averaged 3.1.

According to a survey administered to all participants, 99 percent of the students in both projects had previously worked on some type of group project, and only 1 percent of the students indicated that they had never worked on a team project.

The Turkey-based students were eight hours ahead of the US-based students and seven hours ahead of the Panama-based students, and the Panama-based students were one hour ahead of the US students.

3.3. Team Composition

US, Turkish, and Panamanian students were grouped together for both tasks. Unfortunately, the actual number of students per group often varied according to the class sizes for a particular course. The first task-project had 10 teams, with approximately 3 students in each group from each of three universities (for a total of 9 team members in each group). The second task-project had 15 teams, with between 5-6 students in each group. Each team consisted of approximately 2 students from the US, 2-3 from Turkey, and 1 student from Panama, with a few teams without any Panamanian students.

The students in each task-project team were randomly assigned to their teams. The students were not allowed to change their teams during the project. The language for communication within the project teams was English.

3.4. Tasks

Each of the two tasks had their own individual assignment. As previously stated, these assignments were determined by the curriculum of the courses that were involved in the research for that semester. The first collaborative task was assigned to students enrolled in programming and interface design courses in spring 2008. Thus, this particular assignment consisted of a mid-size software development project involving a fictitious user who was requesting an application that could create groups (such as those that were involved in this project). The input for the application was a set of criteria (as specified by the user) and a file containing a list of names of students who were enrolled in a fictitious course. The output for the project was a list of the groups and the students assigned to those groups. Student teams were given four weeks to complete the project.

The second collaborative task was assigned to students enrolled in a database or Java course in fall 2008. These students were given an assignment to design, create and query a database that could maintain a fleet of rental cars. Students were expected to produce an appropriate E-R diagram and test queries for the database as well as develop a Java application that could add and delete data in the database. Student teams in both projects were also responsible for completing several reports and documentation for their systems. These student teams were also given four weeks to complete their projects.

4. Measures

4.1. Performance Measures

Team and individual scores were obtained through an evaluation of the artifacts delivered by each individual and group. Each deliverable was evaluated based on four criteria – accuracy, efficiency, thoroughness, and style. A design or a program was considered accurate if it satisfied the user's functional specifications and contained no errors. A program's efficiency score was obtained by examining the number and type of program modules included in the final project. A program's thoroughness was scored on whether the design or program included all the necessary elements. Finally, a program's style score was obtained by looking at different programming elements such as variable naming conventions, indentation, use of documentation, etc. Researchers from each university graded their own student projects as well as those from the other participating countries. A mean grade for the project was then assigned to each student. A team's performance was evaluated by averaging the individual grades on each of the assignments.

4.2. Communication Behavior Measures

A survey was administered to team members at the beginning of each task. This survey was designed to collect the demographic information about each student participant. Although the teams for both tasks used a number of different online collaborative tools, they did most of their team communication using an open source platform learning management system called Online Learning and Training (OLAT). This computer managed instructional software supports asynchronous communications such as forums, emails, wikis, file sharing etc., and synchronous communication such as chat. Data from the US-Panama-Turkey projects was obtained from the OLAT system directly, and from programs that were developed to augment OLAT's data collection capabilities. Although the recorded data included information about every communication activity (i.e., message posting, file upload, and wiki entry, along with the date, time, and author of each online activity), this study focused on only the forum data posted by each group.

A content analysis of text was conducted, with a single communication as the unit of analysis. Two trained coders categorized messages into the five content categories using the instrument explained in this section. Each posting was extracted and coded into one of the communication behavior categories: planning, contributing, seeking-input, monitoring/reflecting, and interacting socially. Duplicate codes were assigned whenever an utterance indicated multiple collaborative behaviors. Instructor messages posted by the class instructor or teaching assistant were excluded from the counts. Unclassified messages that did not fit into any of the categories were also not counted. Percent agreement among the two coders for general content was 84.2%.

The instrument that was used to code the group posting was a coding scheme that characterizes a student group's collaborative behaviors [23]. Curtis and Lawson [10] identify nine different behaviors (described in Johnson & Johnson [23]) as being supportive of the collaborative process. Curtis and Lawson first created a set of 15 separate communication activities and then grouped these activities into 5 commuication behavior categories. The list of these behavior categories and their descriptions are given in the Table 2. This instrument was used to determine the extent to which various communication patterns can be used to describe global software development student teams.

The Curtis and Lawson instrument specifies five different levels interactions or behaviors. The *planning* behavior indicates that the message contains a statement that relates to organizing work, initiating activities, or group skills. The contributing code is assigned to messages that gave help, provide feedback, exchange resources, share programming knowledge, challenge others or explain one's position. Other collaborative behaviors are also noted such as seeking input and reflection. Conversations about social matters that are unrelated to the group task at hand are generally placed in the social interaction category.

Table 2. **Coding scheme and communication behavior Categories [30, p.8]**

Behavior Categories	Behaviors
Planning	Group Skills, GS
	Organizing work, OW
	Initiating Activities, IA
Contributing	Help Giving, HeG
	Feedback Giving, FBG
	Exchanging Resources and Information, RI
	Sharing Knowledge, SK
	Challenging others, Ch
	Explaining or elaborating, Ex

Seeking Input	Help Seeking, HeS
	Feedback Seeking, FBS
	Advocating Effort, Ef
Reflection/Monitoring	Monitoring Group Effort, ME
	Reflecting on medium, RM
Social Interaction	Social Interaction, SI

Using these five categories, the authors coded the 561 student messages that had been recorded for the two different tasks. Instructor messages were not included in these message counts.

5. Results

5.1. Summary Data Results

A total of 301 messages were coded for the first project, and 260 messages were coded for the second project. Table 2 lists the project grades for each team and project, with highest grade first.

Table 3. Listing of Grades and Behavioral Communication Activities by Task Type

Task 1 Grades		
Groups	Grades	Activities
5	87	38
7	76	27
2	71	41
10	71	26
9	70	30
3	70	43
6	66	21
8	64	15
1	61	32
4	59	28
Task 2 Grades		
A	89	51
O	79	17
K	75	78
E	74	11
L	73	1
M	71	8
I	69	11
C	64	18
G	64	15
J	64	18
B	59	21

D	57	0
F	56	0
H	55	1
N	54	10

As stated above, previous literature has shown a relationship between the total amount of communication messages and group performance [11]. It was believed that frequent communications would increase a team's information exchange and thus increase team performance. In this study we tabulated the total number of communication behaviors for each team for each task and then correlated the number of communication behaviors with the group grades on the individual projects. There was not a statistically significant correlation between grades and number of communication behaviors for groups assigned the first task *(r = - 0.44, p = .19)*, whereas there was a correlation between grades and number of communication activities for groups assigned to the second task *($r = 0.55$, $p < .05$).*

We also looked at the effect of GPA on group performance to make sure that this variable did not interfere with data concerning the number or kind of communication behaviours that occurred with the groups. GPAs were obtained for 65 of the 85 students assigned to the first task-project, and 63 of the 70 students assigned to the second task-project. There was no correlation between GPA and group performance for teams in the first task-project *($r = 0.123$)*, and only a relatively weak relationship between the variables for groups in the second task-project *($r = 0.41$, $p < .05$).*

Finally, we looked at whether group grades for the two projects differed significantly. An independent *t*-test was used to determine whether there was a difference in group grades between teams assigned to the first task versus those assigned to the second. The results of the t-test revealed that there were no significant differences between the mean grades for teams associated with either task (*Task 1 = 69.4; Task 2 = 66.0); $t = 0.52$, $df = 23$; $p = 0.51$).*

Table 4. Comparison of Mean Scores on Projects for Each Task

Groups	*N*	*Mean*	*SD*
Task 1	10	69.4	8.04
Task 2	15	66.9	2.54

5.2. Communication Patterns and Task Type

Looking beyond the summary data, the authors examined the communication behaviors for each task. Table 5 summarizes the analysis of the collaborative behaviors, as defined by the Curtis and Lawson coding scheme, which took place in the online forums for each of the tasks. Figure 1 represents this same data in a graphical format.

It seems obvious from looking at both the table and the figure that the communication behaviors for the two tasks are surprisingly similar. The largest number of communication behaviors for both tasks occurs in the *contributing* category (42.85% for Task 1; 46.92% for Task 2), and the least numbers occur in the reflection/monitoring and social interaction categories.

Table 5. Communication Behaviors for Groups in Task 1 and Task 2

	Task 1			**Task 2**	
Categories	*# utterance*	*Percentages*		*# utterance*	*Percentages*
Planning	60	19.93		58	22.31
Contributing	129	42.85		122	46.92
Seeking Input	93	30.80		70	23.97
Reflection/Monitoring	9	2.99		9	3.46
Social Interaction	10	3.32		1	0.39
total	301			260	

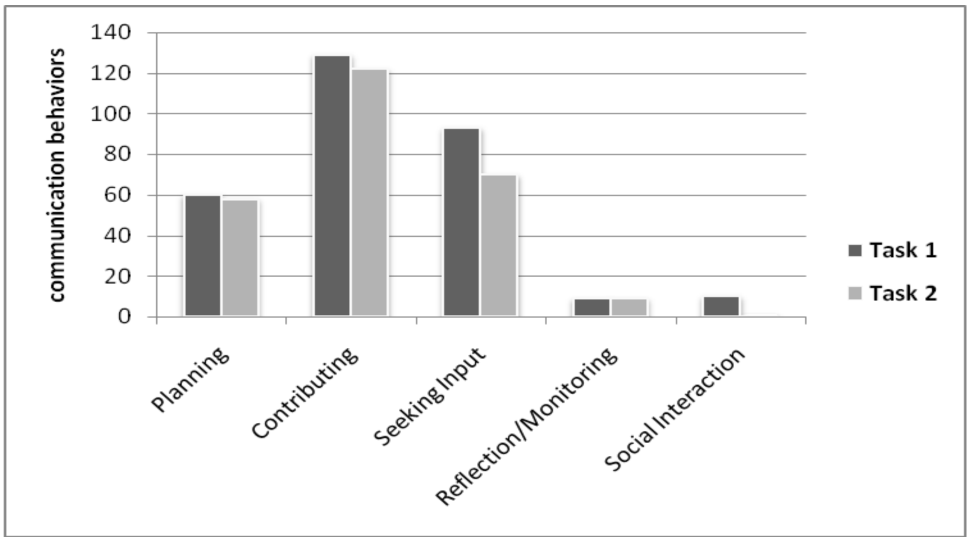

Figure 1. Comparison of Communication Behaviors for each Task

To clarify the comparisons between the communication behaviors that occurred in each task, we converted the raw data into percentages; that is, the total number of utterances in a category over the total number of utterances for that group (Figure 2). A chi-square was carried out to determine whether the communication behaviors that occurred within each task were significantly different. Although there was a higher proportion of social interaction behaviors that occurred in Task 1 versus Task 2 (3.32% versus .39%), the overall differences in the distribution of communication behaviors among the two tasks is very small, χ^2 *(df= 4) = 7.88, p = 0.096).* When the social interaction category (because of small cell size) is removed from the data set, then the p-value indicates that there is insufficient evidence to reject the null hypothesis that the two tasks are significantly different with respect to communication patterns, χ^2 *(df= 3) = 1.62, p = 0.66.* After removing the social interaction category, the two tasks exhibit nearly identical communication behavior patterns: *planning* = 21 versus 22; *contributing* = 44 versus 47; *seeking* = 32 versus 27; *reflecting* = 3 versus 3).

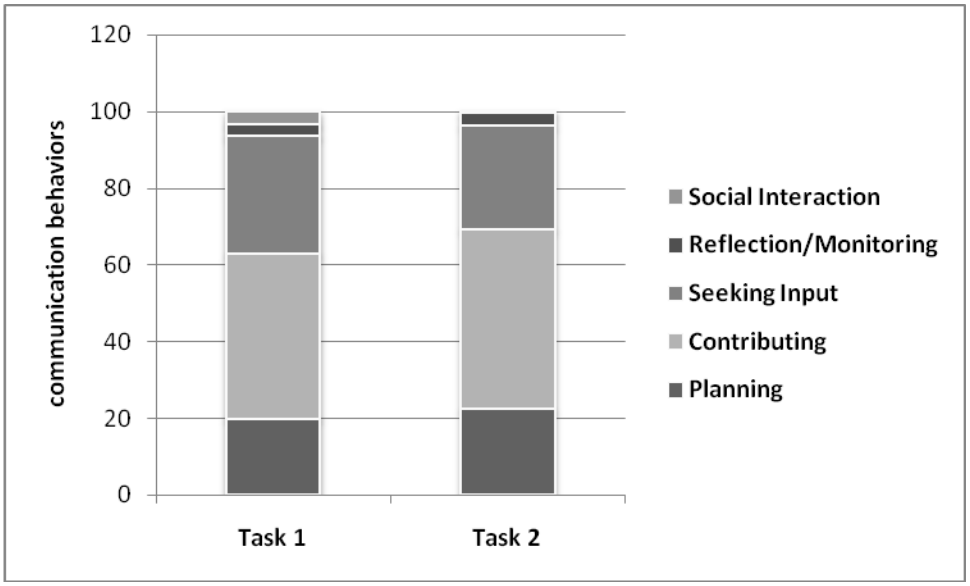

Figure 2. Proportions of Communication Behaviors in Task 1 and Task 2

5.2.1. High versus Low Performing Groups

Since one of the broader goals of the three-year research project is to find ways to improve collaboration among global software teams, we also examined differences in the amount and type of communication behaviors between the high and low performing teams assigned to each of the two tasks. In order to answer this question we selected the two highest and two lowest rated teams in each of the two tasks. As a result of this process, we selected groups 5 and 7 from Task 1 and groups A and O from Task 2 as the high performing teams. The low performing teams were groups 1 and 4 in the first task and groups H and N in the second task.

Figure 3 provides a comparison of the data on the high versus low performing teams for each task. The chi-square test for homogeneity for each task set shows that the overall distributions of communication activities between high and low was significantly different for both tasks (Task 1: [χ^2 *(4, N = 230) = 9.74, p=0.04, r=-0.18, z=-2.78, p=0.002],* and Task 2: χ^2 *(4, N = 155) = 10.7074, p=0.03, r = -0.19, z =-2.32, p = 0.01].* High performers in Tasks 1 and 2 show a much higher proportion of contributing behaviors (43% in Task 1 and 55% in task 2) versus the low performers, while the proportion of planning behaviors seems larger in the low performing teams

Having established the differences between the high and low performing groups, we then compared the distributions of communication activities among the high performers in each task type. Figure 4 shows the percentage of communication behaviors as a function of the task and performance for the high performance group. No pattern difference is apparent in this comparison. A chi-square on the high performing teams' data shows that the distribution of communication activities that occurred in the high performing teams for both tasks are similar, χ^2 *(4, N = 257) = 5.3898, p = 0.2496.* A comparison of the distributions of communication activities of low performing teams in each task produced comparable results, χ^2 *(4, N = 129) = 0.7223, p = 0.9485.* The distribution of communication activities of low performing groups assigned to Tasks 1 and Task 2 did not differ significantly with respect to task type (Figure 5). Thus, both H_2 and H_3 are rejected.

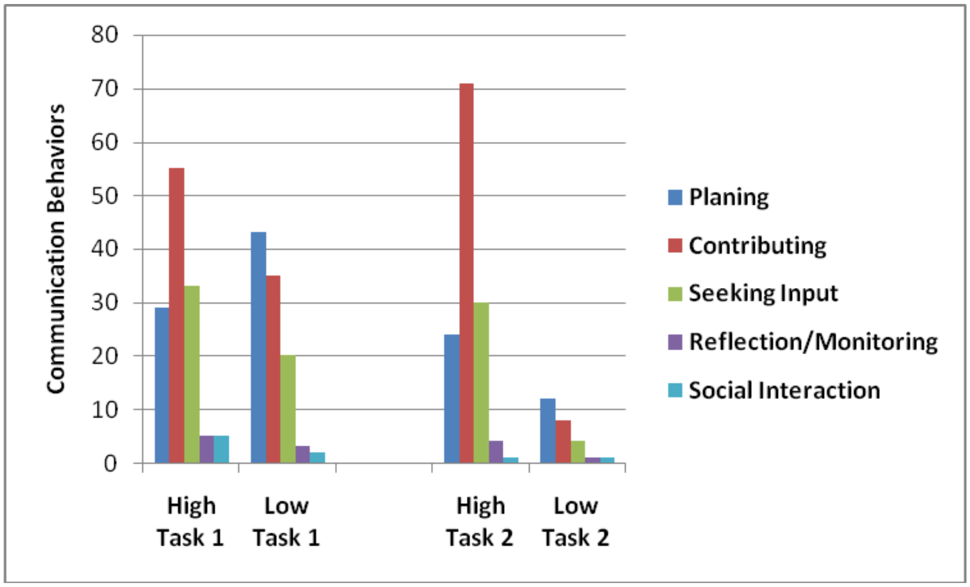

Figure 3. Comparison of Communication Behaviors between High and Low Performing Groups in Task 1 and Task 2

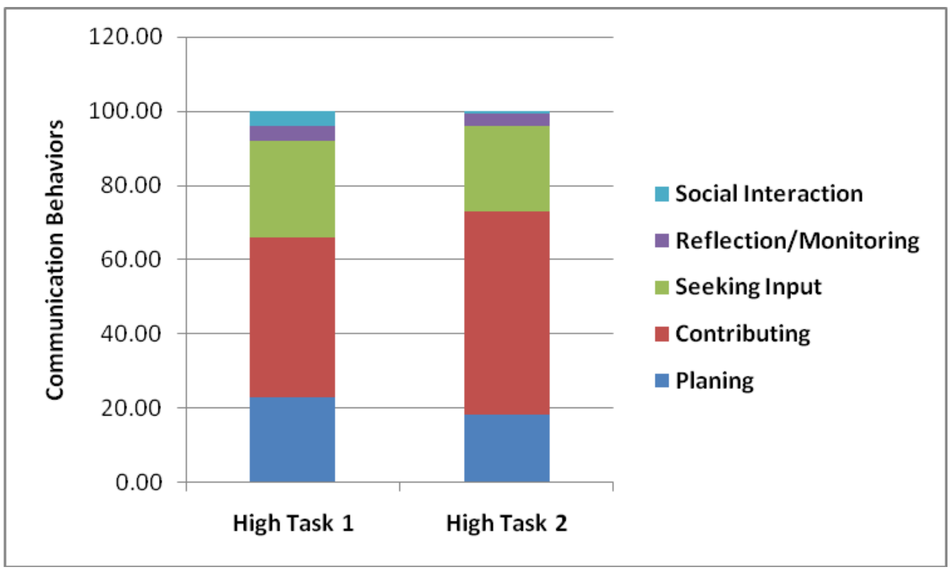

Figure 4. Comparison of Proportions of Communication Behaviors for High Performers in Task 1 and Task 2

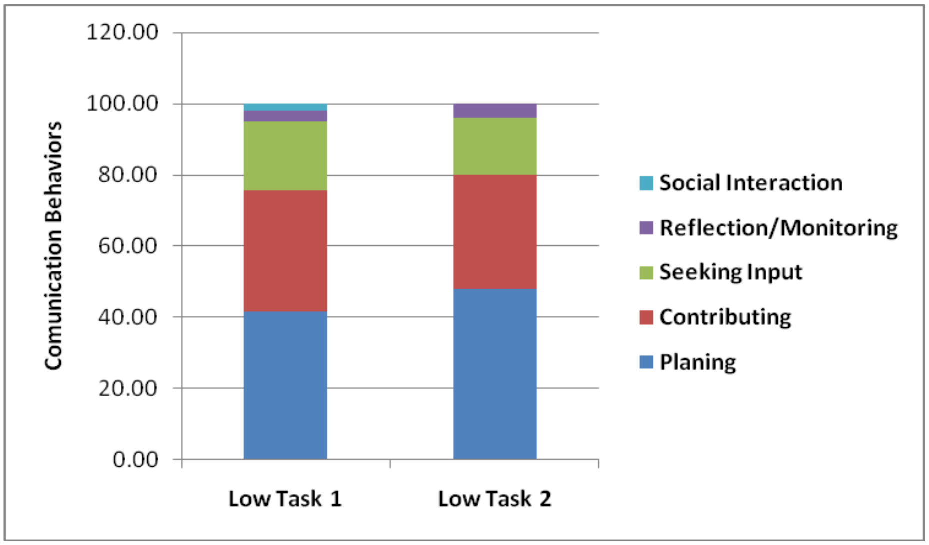

Figure 5. Comparison of Proportions of Communication Behaviors for Low Performers in Task 1 and Task 2

6. Conclusion

A comparison of the online communication behaviors for students engaged in global software development projects has produced some interesting results. The communication behaviors, as described in Curtis and Lawson, provided a way to capture patterns among the participating groups and compare them across different tasks. Among our more significant findings are:

- The frequency of reflecting/monitoring communication behaviors appears to be relatively low regardless of task type. Only about 3 per cent of all messages in Tasks 1 and 2 were coded as being in this category, indicating a low level of high-order processing in online discussions of global software development students. The results of this study add to the growing evidence that students use discussion forums more for exchanging information and reinforcing beliefs rather than deliberating over new ideas and concepts.

- The results of this study do not support hypotheses H_1. Very little difference was observed in the types of messages posted by groups in either of the two tasks. Both groups had a higher proportion of contributing messages, with a smaller percentage of messages in the other four categories. The very small differences in the distribution of messages coded for the experimental conditions suggest that teachers of global software development courses may have a model of communication patterns that can be used to help students complete any distributed group programming task. Specification of this model may help reduce the number of irrelevant messages, as students learn how to spend more time and effort in more productive communication activities. Such student attention on specific communication activities may be beneficial to the goal of increased performance in global software development projects.

- However, the data presented in this paper may be insufficient to establish or reject a relationship between task context and communication behaviors. Our previous research [37] suggests that there are such differences, and other studies have reported similar results. The earlier study found that groups assigned to produce a database displayed significantly more planning and less contributing behaviors than the groups assigned to complete a programming task. Since the data for this study suggest that task context has no large effect on communication behaviors, there is a need to resolve the inconsistencies. A comparison of the two studies indicates that the database task assigned to groups in the first research project focused on only the design part of the database, while the database task in the current study focused on both the design and implementation of a database along with its interface. This suggests that further research is needed to discover communication behavior differences between requirements and implementation student tasks.

- There was a significant relationship between number of messages posted and grade on the group projects for both tasks. However, the number of messages posted is a rough indicator of discussion quality, and does not necessarily relate to the actual production of a product. Thus, we examined the communication differences between high and low performing teams and discovered that high performing teams spent a higher percentage of their time contributing to the overall completion of the task, while low performing teams tended to spend their time planning or seeking information. After this analysis, we examined the distributions of communication behaviors among high performing teams for the two tasks and found that the differences were not significant. We also examined the communication behaviors among low performers and also found that the differences in the distributions were not significant. Thus task context seems to have no large effect on the communication behaviors of either high or low performers, resulting in a rejection of H_2 and H_3.

To improve the performance of global software student projects, researchers need to determine if the behaviors discussed in this paper consistently lead to higher performance among teams. This study did not find evidence that task type affects overall patterns of communication behaviors, nor was task type found to affect the communication patterns of high and low performers. Instructors of global software development courses may be able to use the communication behavior model in this study as a way to promote more effective group processes within distributed team projects. This study can also be used to help educators develop strategies that can maximize the various factors that characterize more successful collaborations.

7. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0705638. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We wish to thank the students who participated in the study, and the many colleagues at each institution who helped make this research possible.

References

1. J. Andriessen, Collaboration in computer conferencing. In A. M. O'Donnell, C. E. Hmelo-Silver, & G. Erkens (Eds.). Collaborative learning, reasoning, and technology, 2006, 197–232. Mahwah, New Jersey: Lawrence Erlbaum Associates.

2. K. Agerup, M. Busser, A case-study on collaborative learning in distributed, cross-cultural teams, International Conference on Engineering Education, Gainesville, Fl., October 16-21, 2004.

3. R. Bakeman, J. Gottman, Observing interaction: An introduction to sequential analysis, Cambridge, MA: Cambridge University Press, 1997.

4. P. Bouillon, J. Krinke, S. Lukosch, Software engineering projects in distant teaching, Proceedings of Conference on Software Engineering & Training, 2005, 147-154.

5. L. Burnell, J. Priest, J. Durrett, Teaching distributed multidisciplinary software development, IEEE Software, 29 (5) (2002) 86-93.

6. E. Carmel, Global Software Development Teams: Collaborating Across Borders Time Zones. Prentice Hall, Upper Saddle River, NJ, 2002.

7. K. Cho, D. Jonassen, The effects of argumentation scaffolds on argumentation and problem solving, Educational Technology: Research & Development, 50 (3) (2002) 5-22.

8. C, Cramton, P. Hinds, Subgroup dynamics in internationally distributed teams: Ethnocentrism or cross-national learning? Research in Organizational Behavior, 26 (2005) 231-263.

9. I Crnkovic, R. Land, Taking global software development from Industry University and back again, Proceedings of ICSE International Workshop on Global Software Development, Sjogren, 2003J, 2003.

10. D. Curtis, M. Lawson, Exploring Collaborative On-line Learning, JALN 5 (1) (2001) 21-34.

11. J. Cushing, K. Cunningham. G. Freeman, Towards best practices in software teamwork, Journal of Computing Sciences in Colleges 19 (2) (2004) 72 – 81.

12. B. De Wever, T. Schellens, M. Valcke, H. Van Keer, Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. Computers and Education, 46, (2006) 6–2

13. H. Edwards, V. Sridhar, Analysis of the effectiveness of global virtual teams in software engineering projects, Proceedings of the 36th Hawaii International conference on systems sciences (HICSS03), hicss 19b, 2003.

14. C. Fulford, S. Zhang, Perceptions of interaction: The critical predictor in distance education, The American Journal of Distance Education 7 (3), (1993) 8-21.

15. D. Garrison, T. Anderson, W. Archer, Critical thinking, cognitive presence, and computer conferencing in distance education. American Journal of Distance Education 15(1) (2001) 7-23.

16. M. Guzdial, P. Ludovice, M. Realff, T. Morley, K. Carroll, When collaboration doesn't work, Proceedings of the International Conference of the Learning Sciences, 2002, 125-130.

17. J. Herbsleb, A. Mockus, T. Finholt, R. Grinter, Distance, dependencies, and delay in a global collaboration, Proceedings of the ACM Conference on Computer Supported Cooperative Work, 2000, 319-328.

18. P. Herder, E. Sjoer, Group-based learning in internationally distributed teams: An evaluation of a cross-Atlantic experiment, ASEE/IEEE Frontiers in Education Conference, Boulder, CO., S1F_7-S1F_12, 2003.

19. J. Hewett, How habitual online practices affect the development of asynchronous discussion threads. Journal Education Computer Research 28 (2003) 31-45.

20. A. Hollingshead, G. Wittenbaum, G. Jacobsohn, S. Faidin, Competitive members in cooperative groups, JACM Conference, 2005.

21. A. Hollingshead, J. McGrath, K. O'Connor, Group task performance and communication technology, Small Group Research, 24 (3) (1993) 307-333.

22. A. Jeong, Sequential analysis of group interaction and critical thinking in online threaded discussions, The American Journal of Distance Education, 17 (1) (2003) 25-43.

23. D. Johnson, R. Johnson, Cooperation and the Use of Technology. In D. H. Jonassen (Ed.), Handbook of research for educational communications and technology, 1017-1043, 1996, New York: Macmillan.

24. R. Jorczak, The effects of task characteristics on higher-order learning in online collaborative learning. Unpublished doctoral dissertation. Minneapolis, MN: University of Minnesota, 2008.

25. R. Kraut, L. Streeter, Coordination in software development, Communication of the ACM, 38 (3) (1995) 69–81.

26. P. Layzell, O. Brereton, A. French, Supporting collaboration in distributed software engineering teams, The Asia-Pacific Software Engineering Conference, December 5-8, 2000, 38-45.

27. R. Marra, A review of research methods for assessing the content of computer-mediated discussion, Journal of Interactive Learning Research, 17 (3) (2006) 243-267.

28. R. Marra, J. Moore, A. Klimczek, A Comparative analysis of content analysis protocols for online discussion forums, Educational Technology Research and Development 52 (2) (2004) 23 - 40.

29. B. Meyer, The unspoken revolution is software engineer, IEEE Computer, 39 (1) (2006) 121-123.

30. B. Munkvold, L. Line, Training students in distributed collaboration: Experiences from two pilot projects, Journal of Informatics Education and Research, 3 (2) (2005) 1-17.

31. D. Newman, C. Johnson, C. Cochrane, B. Webb, An experiment in group learning technology, Interpersonal Computing and Technology, 4 (1) (1996) 57-74.

32. M. Paasivaara, C. Lassenius, Collaboration practices in global inter-organizational software development projects, Journal Software Process: Improvement and Practice, 8(4) (2003) 183-199.

33. M. Purvis, S. Cranfield, Educational experiences from a global software engineering project, 6th Australasian Computing Education Conference (ACE2004), Dunedin, New Zealand, 269-275, 2004.

34. S. Sarker, S. Sahay, Understanding virtual team development: An interpretive study, Journal of Association Information Science, 4(1) (2003).

35. V. Savicki, M. Kelley, D. Lingenfelter, Gender, group composition, and task type in small task groups using computer-mediated, Computers in Human Behavior, 12 (4) (1996) 549-565.

36. M. Scardamalia, C. Bereiter, Computer support for knowledge-building communities, The Journal of the Learning Sciences, 3 (3) 1994, 265-283.

37. F. Serce, K. Swigger, F. Alpaslan, R. Brazile, G. Dafoulas, V. Lopez, Exploring the communication behavior among global software development learners, International journal of Computer Applications in Technology, (in press).

38. A. Sherry, C. Fulford, S. Zhang, Assessing distance learners' satisfaction with instruction. The American Journal of Distance Education, 12 (3) (2004) 4-28.

39. D. Smite, Requirements management in distribute projects, Journal Universal Knowledge Management, 1 (2) (2006) 69-76.

40. T. Schümmer, GAMA - A Pattern Language for Computer Supported Dynamic Collaboration," Proceedings of the 8th European conference on pattern languages of programs, 2004, 53-11, Konstanz: UK.

41. J. Strijbos, P. Kirschner, R. Martens, What we know about CSCL: And implementing it in higher education, Boston, MA: Kluwer Academic/Spinger Verlag, 2004.

42. A. Veerman, E. Veldhuis-Diermanse, Collaborative learning through electronic knowledge construction. In A. M. O'Donnel, C. E. Hmelo-Silver, & G. Erkens (Eds.). Collaborative learning, reasoning, and technology 323–354, 2006. Mahwah, New Jersey: Lawrence Erlbaum Associates.

43. G. von Krogh, J. Ross, Kleine, Knowing in Firms: Understanding, Managing and Measuring Knowledge, Sage Press, London, 1998.

44. J. Walther, Computer-mediated Communication: Impersonal, interpersonal, and hyperpersonal interaction, Communication Research, 23 (1) (1996) 3-43.

45. J. Wiley, J. Bailey, (2006). Effects of collaboration and argumentation on learning from web pages. In A. M. O'Donnel, C. E. Hmelo-Silver, & G. Erkens (Eds.). Collaborative learning, reasoning, and technology, 2006, 297–322. Mahwah, New Jersey: Lawrence Erlbaum Associates.